**API DOCUMENTATION**

Folder Tree (tf-serving):

* nginx/ → webserver and load balancer
* tf-serve/ → Tensorflow serving for Yolov4 and SSD model
* web/ → Flask API
* test/ → for testing purposes
* docker-compose.yml

Requirements:

* docker
* docker-compose
* curl

Running program using:

$ sudo docker-compose up --build -d

In new terminal, go to folder test, and run:

$ sh test\_curl.sh

Or

$ python3 simple\_request.py

| BASE\_URL | localhost:8080 |
| --- | --- |
| ENDPOINT | https://BASE\_URL/predict |
| Description | Run prediction for given image |
| Method | POST |
| Body | image → image file  method → choice either “yolov4” or “ssd”  margin → padding for cropped detection results |
| Path Parameter | - |
| Example Request | curl -X POST -F image=@image.png -F method=yolov4 -F margin=25 http://localhost:8080/predict |
| Example Response | **Success**:  {  "results":[  {  "bboxes":[  [179,89,237,152],[154,372,227,450]  ],  "classes":[1,1],  "confidences":[0.993786156,0.988828361],  "crop\_results":["base64image","base64image"],  "img\_output":"base64image",  "margin":"25",  "method":"yolov4",  "num\_detections":2  }  ],  "success":true  } |
| Response Notes | bboxes → array of bounding boxes in [x1,y1,x2,y2] where (x1,y1) is top left and (x2,y2) is bottom right  classes → array of classes  confidence → array of confidence  crop\_results → array of cropped detection (in base64 image forms)  img\_output → image output with detection (in base64 image form)  margin → used padding for cropped detection results  method → used method for detection  num\_detections → number of detected objects |